计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.190500089

基于 ECOC 的多类代价敏感分类方法

吴崇明[1]　王晓丹[2]　薛爱军[2]　来　杰[2]
1 西京学院商学院　西安 710123
2 空军工程大学防空反导学院　西安 710051
(w_9887@163.com)

摘　要　研究了基于纠错输出编码实现多类代价敏感分类的方法,提出了一种新的将多类代价敏感分类问题分解为多个二类代价敏感分类问题的框架。为获得其中每个二类代价敏感基分类器的二类代价矩阵,提出了利用已知多类代价矩阵计算误分类代价的期望值的方法,给出了计算二类代价矩阵的通用计算公式。为验证所提方法的有效性,在人工和 UCI 数据集上将其与现有方法进行了比较,实验结果表明所提方法具有相似甚至更好的性能。

关键词:多类代价敏感分类;纠错输出编码;多类代价矩阵;二类代价矩阵
中图法分类号　TP391

Multiclass Cost-sensitive Classification Based on Error Correcting Output Codes

WU Chong-ming[1],WANG Xiao-dan[2],XUE Ai-jun[2] and LAI Jie[2]
1 Business School,Xijing University,Xi'an 710123,China
2 College of Air and Missile Defense,Air force Engineering University,Xi'an 710051,China

Abstract　Approach of multiclass cost-sensitive classification based on error correcting output codes is studied in this paper,and a new framework to decompose the complex multiclass cost-sensitive classification problem into a series of binary cost-sensitive classification problems is proposed. In order to obtain the binary cost matrix of each binary cost-sensitive base classifier,a method of computing the expected misclassification costs from the given multiclass cost matrix is proposed,and the general formula for computing the binary costs are given. Experimental results on artificial datasets and UCI datasets show that the proposed method has similar or even better performance in comparison with the existing methods.

Keywords　Multiclass cost-sensitive classification,Error correcting output codes,Multiclass cost matrix,Binary cost matrix

1　引言

对许多实际应用问题,如疾病诊断[1]和欺诈检测[2]等,不同类型的分类错误造成的误分代价明显不同,这些都属于代价敏感分类问题。为解决代价敏感分类问题,研究者提出了多种方法[1-9]。根据实现策略的不同,代价敏感学习主要分为两大类:间接代价敏感学习(或代价敏感元学习)[1,4]和直接代价敏感学习[5-7]。间接代价敏感学习并不改变传统的学习算法,而是将代价敏感学习方法作为一个单独的模块,对训练数据进行预处理或者对学习算法的输出结果进行后处理,从而实现代价敏感学习。直接代价敏感学习是直接将分类代价引入到分类器的构建中,通过更改分类器的结构或者修改分类器的目标函数,将传统的以最小错误率为目标的分类器转换为以最小风险为目标,从而实现代价敏感学习。

对于如何实现多类代价敏感分类,将纠错输出编码(Error-Correcting Output Codes,ECOC)[10-12]多类分解框架与代价敏感二分类器结合引起了研究者的关注。文献[14]将多类代价敏感分类问题分解为一系列二分类代价敏感分类问题,通过集成方法由各二分类代价敏感基分类器的输出生成多类分类结果。Lin[15]利用代价转换技术将"一对一"和"一对多"编码方法扩展到了多类代价敏感分类问题。与 Lin 的工作相比,Langford[16]提出了一个更加通用的框架,该框架基于修改后的 ECOC,将多类代价敏感分类问题分解为若干个二类代价敏感分类问题。但是,该框架的性能较差,其中包含的取阈值的步骤,使得代价信息的表达变得不准确[17]。如何设计更有效、更通用的多类代价敏感分类方法仍是具有挑战性的问题。

基于 ECOC 多类分解框架实现多类代价敏感分类方法,通过 ECOC 多类分解框架将多类代价敏感分类问题分解为一系列二类代价敏感分类问题,存在的一个关键问题:如何从给定的多类代价矩阵出发得到每个类的错分代价,并设计新的、更加有效的分解框架;如何在将多类代价敏感分类问题分解为一系列二类分类代价敏感问题的同时,由给定的多类代价矩阵得出对应的一系列二类代价矩阵。本文基于原始 ECOC 框架,提出了一种新的分解框架;为获得其中每个二类代价敏感基分类器的二类代价矩阵,提出了利用已知多类代价矩阵计算误分类代价的期望值的方法;最后给出了计算二类代价矩阵的通用计算公式。

基金项目:国家自然科学基金(61876189,61273275,61703426)
This work was supported by the National Natural Science Foundation of China(61876189,61273275,61703426).
通信作者:王晓丹(afeu_w@163.com)

2 二类代价敏感纠错输出编码

纠错输出编码基于二元{1,−1}或三元{1,0,−1}的编码矩阵来实现类别的分解和基分类器输出结果的融合。{1}和{−1}分别代表一类子类,{0}代表对应的类别不参与相应基分类器的生成。图 1 给出了一个 4 类样本数据的编码矩阵示意图。

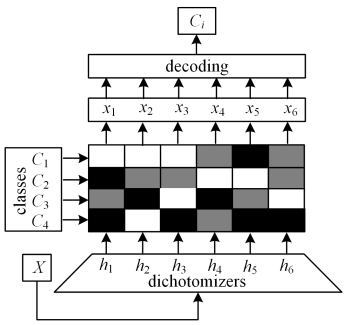

图 1 4 类模式 ECOC 分类

Fig. 1 4 classes ECOC classification

编码矩阵中行代表类别,列代表基分类器。原始类别通过每一列对应的二类划分分成正、负子类,分别用 1 和 −1 表示,即白色和黑色,灰色代表码字 0。在获得二类划分之后,用新的正负子集训练基分类器,并对待识别样本进行测试,输出结果,得到输出向量(结果可以为软输出也可以为硬输出),最后确定解码策略进行基分类器输出结果的融合。

常见的“一对一”和“一对多”多类分类方法属于 ECOC 分解框架的两种特殊情况,即使是层次分解、决策树分解,也能得到对应的 ECOC 编码矩阵。ECOC 方法简单有效,避免了对复杂多类分类问题直接进行建模,使多类分类能够利用二分类中经典的成熟理论和方法,同时继承了纠错输出编码的容错能力[10]。

对于代价敏感多类分类问题,多类代价矩阵(Multiclass Cost Matrix,MCM)表示了各类不同的误分类的代价,例如,4 类代价矩阵如式(1)所示:

$$\begin{bmatrix} \omega_{11} & \omega_{12} & \omega_{13} & \omega_{14} \\ \omega_{21} & \omega_{22} & \omega_{23} & \omega_{24} \\ \omega_{31} & \omega_{32} & \omega_{33} & \omega_{44} \\ \omega_{41} & \omega_{42} & \omega_{43} & \omega_{44} \end{bmatrix} \tag{1}$$

其中,ω_{ij} 表示将一个属于 c_i 类的样本错分为属于 c_j 类的代价或风险。

对于代价敏感多类分类问题,也可按照 ECOC 多类分解框架实现,但需要同时考虑多类代价矩阵,以及分解后每个二类代价敏感基分类器的二类代价矩阵,设计分类器的目的是最小化误分类代价。基于上述考虑,本文给出一个新的 ECOC 分解框架,称之为二类代价敏感纠错输出编码(Binary Cost Sensitive Error Correcting Output Codes,BCS_ECOC)。图 2 给出了 4 类 BCS_ECOC 的一个示例。

BCS_ECOC 编码矩阵的定义与 ECOC 中编码矩阵的定义相同。同时,BCS_ECOC 也包含两个重要的阶段:训练阶段和测试阶段。在训练阶段,训练数据集被重组为两个超类:正类和负类。这一过程根据编码矩阵每一列元素的符号完成。然后,基于各二类分类基分类器的代价矩阵,二类代价敏感基分类器在重组后的训练数据集上完成训练。在测试阶段,利用训练得到的代价敏感基分类器对未知类别的样本进行分类,得到的分类结果将根据解码规则进行解码,这与 ECOC 中的相同。

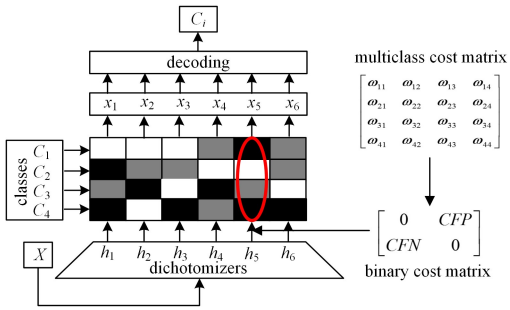

图 2 BCS_ECOC 的一个示例

Fig. 2 Example of BCS_ECOC

本文二类代价敏感纠错输出编码 BCS_ECOC 与传统纠错输出编码 ECOC 的不同体现在:1)设计目标不同,ECOC 的设计目标是最小化错误分类率,BCS_ECOC 的设计目标是最小化误分类代价;2)基分类器不同,传统 ECOC 中的基分类器是非代价敏感的二类分类器,如支持向量机,BCS_ECOC 中的基分类器则是代价敏感的二类分类器,如代价敏感支持向量机。基于 BCS_ECOC,许多成熟的二类代价敏感分类器,都可被扩展用于解决多类代价敏感分类问题。

在上述 BCS_ECOC 框架中,需要解决的一个关键问题是:如何基于已知的多类代价矩阵为每个二类代价敏感基分类器计算对应的二类代价矩阵。下节将给出解决。

3 基分类器二类代价矩阵的生成

如何根据已知的多类代价矩阵为每个二类代价敏感基分类器计算二类代价矩阵,是本节要解决的问题。以下首先探讨当编码矩阵为“一对一”编码时二类代价矩阵的计算方法,然后探讨当编码矩阵为“一对多”编码时二类代价矩阵的计算方法,最后给出基于多类代价矩阵计算二类代价矩阵的通用计算公式。

3.1 “一对一”编码的二类代价矩阵计算

当编码矩阵为“一对一”时,为每个二类代价敏感基分类器计算二类代价矩阵相对比较容易,因为这种情况下(训练基分类器的)正类和负类都只包含一个类。图 3 给出了一个“一对一”编码的例子,其中包含的类别个数为 4。在图 3 中,以基分类器 f_1 为示例,其正类(Positive Class)为 c_1,负类(Negative Class)为 c_2;对于二类代价敏感基分类器 f_1,需要确定如式(2)所示的代价矩阵。

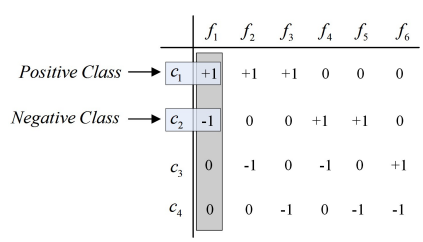

图 3 “一对一”编码某一列对应的正类和负类

Fig. 3 Two classes in one of dichotomizers induced by one-to-one coding matrix

$$\begin{array}{c} \\ P \\ N \end{array}\begin{array}{c} \begin{array}{cc} P & N \end{array} \\ \begin{bmatrix} 0 & CFN \\ CFP & 0 \end{bmatrix} \end{array} \tag{2}$$

其中，CFN 为错分为负类的代价(Cost for a False Negative，CFN)，即正类误分类的代价，此处为属于正类的样本被错分为属于负类的代价；CFP 为错分为正类的代价(Cost for a False Positive，CFP)，即负类误分类的代价，此处为属于负类的样本被错分为属于正类的代价。对于要分类 c_1 和 c_2 这两个类的基分类器 f_1，其代价矩阵中的这两个错分代价 CFN 和 CFP 都可以从给定的多类代价矩阵中直接获得。

CFN 就是将一个属于类 c_1 的样本错分为属于类 c_2 的代价。在如式(1)所示的多类代价矩阵中，$CFN=\omega_{12}$。另一方面，CFP 是将一个属于类 c_2 的样本错分为属于类 c_1 的代价，在多类代价矩阵中，$CFP=\omega_{21}$。所以，基分类器 f_1 的二类代价矩阵如式(3)所示。同理，可以得到其他基分类器的二类代价矩阵。

$$\begin{array}{c} \\ P \\ N \end{array}\begin{array}{c} \begin{array}{cc} P & N \end{array} \\ \begin{bmatrix} \omega_{11} & \omega_{12} \\ \omega_{21} & \omega_{22} \end{bmatrix} \end{array} \tag{3}$$

3.2 “一对多”及其他编码方法的二类代价矩阵计算

对于“一对多”及其他编码矩阵来说，计算某一个基分类器的二类代价矩阵相对比较困难。其原因是，此时在二类代价敏感分类问题中，正类和负类通常会包含多个原始类。图 4 给出了一个“一对多”编码矩阵的例子，它包含的类别数为 4。

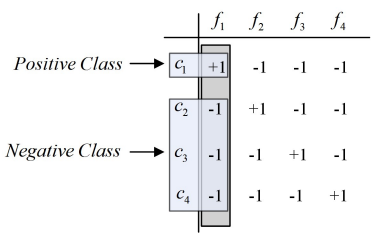

图 4 “一对多”编码中基分类器对应的正类和负类

Fig. 4 Two classes in one of dichotomizers induced by one-to-many coding matrix

在图 4 中，仍然以基分类器 f_1 作为示例。可以看出，对于 f_1，正类仅由类 c_1 组成，而负类由 3 个类 c_2，c_3 和 c_4 组成。此时，错分为负类的代价 CFN 包含了 3 个方面：将属于类 c_1 的样本错分为属于类 c_2 的代价，将属于类 c_1 的样本错分为属于类 c_3 的代价，将属于类 c_1 的样本错分为属于类 c_4 的代价。另一方面，错分为正类的代价 CFP 同样包含 3 个方面：将属于类 c_2 的样本错分为属于类 c_1 的代价，将属于类 c_3 的样本错分为属于类 c_1 的代价，将属于类 c_4 的样本错分为属于类 c_1 的代价。计算基分类器 f_1 的二类代价矩阵，关键在于如何建立 CFN 和 CFP 与这 3 个方面的代价之间的联系。

我们认为，正类误分类的代价 CFN 和负类误分类的代价 CFP 可以通过计算误分类代价的期望值得到。假设正类包含的类的集合为 S_P，负类包含的类的集合为 S_N。对于如图 4 所示的基分类器 f_1，有 $S_P=\{c_1\}$ 和 $S_N=\{c_2,c_3,c_4\}$。以 CFN 的计算为例进行说明，正类误分类的代价 CFN 的计算公式为：

$$CFN=\sum_{c_i\in S_P}P(c_i)\sum_{c_j\in S_N}P(c_i,c_j)\omega(c_i,c_j) \tag{4}$$

其中，$P(c_i)$ 是类 c_i 的先验概率，它满足条件 $\sum_{c_i\in S_P}P(c_i)=1$。$P(c_i,c_j)$ 是属于类 c_i 的样本错分为属于类 c_j 的概率，$\omega(c_i,c_j)$ 是属于类 c_i 的样本错分为属于类 c_j 的代价，它的值可以从多类代价矩阵中直接找到。特别地，需要注意的是概率 $P(c_i,c_j)$ 必须满足条件 $\sum_{c_i\in S_N}P(c_i,c_j)=1,c_i\in S_P$。

同理，负类误分类的代价 CFP 的计算公式为：

$$CFP=\sum_{c_i\in S_N}P(c_i)\sum_{c_i\in S_P}P(c_i,c_j)\omega(c_i,c_j) \tag{5}$$

在式(4)和式(5)中，某一类的先验概率 $P(c_i)$ 可以根据每一类包含样本的个数估计得到，错分代价 $\omega(c_i,c_j)$ 可以在多类代价矩阵中找到。计算 CFN 和 CFP 的困难在于如何得到错分概率 $P(c_i,c_j)$，可以用两类之间的错误分类率近似估计错分概率。

通常，如果两类之间的类别的可分性越好，则分类错误率就越低，因此，错分概率 $P(c_i,c_j)$ 与类别可分性成反比关系。所以，错分概率 $P(c_i,c_j)$ 的计算公式可表示为：

$$P(c_i,c_j)=\frac{1}{J(c_i,c_j)} \tag{6}$$

其中，$J(\cdot,\cdot)$ 为类别可分性的度量准则。常见的类别可分性度量有类内类间距离、Bhattacharyya 距离、Chernoff 界等。

需要注意的是，式(4)和式(5)可以用来为不同种类的编码矩阵计算二类代价矩阵，比如，密集随机编码、稀疏随机编码以及其他基于问题的编码方法等。因此，这两个公式可以作为根据多类代价矩阵计算二类代价矩阵的通用公式。

以下通过实验验证这两个公式的有效性，同时检验 BCS_ECOC 的分类性能。

4 实验及分析

本节从实验数据、实验设计以及实验结果与分析 3 个方面对实验进行介绍。

4.1 实验数据

实验选取了两种不同的数据集，分别是人工数据集和 UCI 数据集。

(1)人工数据集

应用人工数据集的主要优势在于，人工数据集中每一类的条件概率密度函数是已知的，可以更加精确地计算类别之间的可分性。例如，基于每个类已知的条件概率可以计算两类之间的 Bhattacharyya 距离，它可以比类内类间距离更加准确地反映类间的可分性。这些人工数据集共包含 6 类，每一类含有相同数目的样本，其样本分布如图 5 所示，每一类的特征向量的维数为 2，分别用 Feature1 和 Feature2 来表示。每一类的特征向量都服从正态分布，其概率密度函数定义如式(7)所示，其中的参数设置见表 1。

$$p(x|class_i)=\frac{1}{2\pi|\Sigma_i|^{\frac{1}{2}}}\exp\left[-\frac{1}{2}(x-\mu_i)^{\mathrm{T}}\Sigma_i^{-1}(x-\mu_i)\right]$$
$$i=1,2\cdots,6 \tag{7}$$

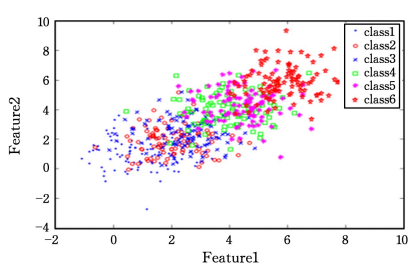

图 5 人工数据集中的数据分布

Fig. 5 Data distribution in artificial datasets

表 1 人工数据集的参数设置

Table 1 Parameters settings for artificial datasets

Class	Prior probabilities	Mean vectors	Covariance matrices
C_1	$P(C_1)=\frac{1}{6}$	$\mu_1=(1,1)^T$	$\Sigma_1=\begin{pmatrix}1&0\\0&1\end{pmatrix}$
C_2	$P(C_2)=\frac{1}{6}$	$\mu_2=(2,2)^T$	$\Sigma_1=\begin{pmatrix}1&0\\0&1\end{pmatrix}$
C_3	$P(C_3)=\frac{1}{6}$	$\mu_3=(2.5,2.5)^T$	$\Sigma_3=\begin{pmatrix}1&0\\0&1\end{pmatrix}$
C_4	$P(C_4)=\frac{1}{6}$	$\mu_4=(4,4)^T$	$\Sigma_4=\begin{pmatrix}1&0\\0&1\end{pmatrix}$
C_5	$P(C_5)=\frac{1}{6}$	$\mu_5=(4.5,4.5)^T$	$\Sigma_2=\begin{pmatrix}1&0\\0&1\end{pmatrix}$
C_6	$P(C_6)=\frac{1}{6}$	$\mu_6=(6,6)^T$	$\Sigma_2=\begin{pmatrix}1&0\\0&1\end{pmatrix}$

在相同的参数下，为了进一步增加实验的鲁棒性，共生成了 6 个具有不同样本数目的人工数据集。这 6 个人工数据集中每类包含的样本的数目分别为 10，50，100，200，300 和 400。为了表述方便，在下文中分别用 1＃，2＃，3＃，4＃，5＃和 6＃来代表这 6 个人工数据集。

(2)UCI 数据集

UCI 数据集[18]常被用来衡量分类方法的性能，实验从 UCI 数据集中选取了 14 个多类的数据集。表 2 列出了这些数据集的样本的数目、特征量的个数以及包含类别的个数。

表 2 实验中用到的 UCI 数据集

Table 2 UCI data sets used in experiment

Dataset	Samples	Attributes	Classes
balance	625	4	3
glass	214	10	6
iris	150	4	3
letter	1214	16	26
page-blocks	5473	10	5
sat	6435	36	6
segmentation	2310	19	7
soybean	306	35	18
thyroid	215	5	3
vehicle	846	18	4
vowel	990	13	11
wine	178	13	3
yeast	1484	8	10
zoo	101	16	7

4.2 实验设计

(1)实验目的及对比方法

实验目的是考查本文提出的二类代价敏感纠错输出编码框架 BCS_ECOC 是否具有与现有方法相似或者更好的分类性能。实验将 BCS_ECOC 在“一对一”编码矩阵和“一对多”编码矩阵下的代价敏感分类方法(分别简称为 BCS-OVO，BCS-OVA)与已有同类方法进行比较。实验比较的方法包括 CSOVO[17]，SECOC-OVO[16]，CSOVA[15]，SECOC- OVA[16]。文献[17]已经将 CSOVO 与其他方法进行了比较，这些方法包括 Weighted All-pairs (WAP)，Tree，Filter Tree (FT)[19]以及 All-pair Filter Tree (APFT)[19]。文献[17]中，通过在大量代价敏感分类数据集上的实验，作者指出 CSOVO 的性能与 WAP 相当，但明显好于其他的几种方法。所以，在实验中，我们关注于 BCS_ECOC 框架下的方法(BCS-OVO，BCS-OVA)是否拥有与 CSOVO 相似甚至更好的性能。此外，对于 CSOVO，CSOVA，SECOC-OVO 和 SECOC-OVA[1)]，本文方法的源代码在此基础上进行了扩展。

(2)实验分组设计

为了消除不同编码矩阵的影响，在每个数据集上将这些方法被划分为了两组：一组是 BC-CSOVO，CSOVO[17]和 SECOC-OVO[16]；另一组是 BC-CSOVA，CSOVA[15]和 SECOC-OVA[16]。对每一组中的方法的性能进行相互比较。同时，引入了 3 种不同类型的多类代价矩阵[20]。对于每个数据集，只随机生成一个多类代价矩阵，以保证所有方法都在同一个多类代价矩阵下进行比较。

(3)实验及参数

对于每一个数据集，利用 10 重交叉验证的结果来衡量每个方法的性能。采用 10 重交叉验证结果的误分类代价的平均值作为每个方法最终的误分类总代价。对于所有方法，二类代价敏感基分类器为代价敏感支持向量机，核函数为高斯径向基核函数，参数全部采用默认设置。需要指出的是，对于本文提出的方法，在人工数据集上以 Bhattacharyya 距离作为类间的可分性度量，在 UCI 数据集上以类内类间距离作为类间的可分性度量。同时，解码方法为最小汉明距离解码。

4.3 实验结果及分析

(1)人工数据集

表 3－表 5 分别列出了 3 种不同类型的多类代价矩阵[20]下，不同方法在人工数据集上的误分类总代价。

表 3 不同方法在人工数据集上的误分类总代价(第一种类型多类代价矩阵)

Table 3 Test cost of different methods on artificial datasets using cost matrix in type a

Datasets	BCS-OVO	CSOVO	SECOC-OVO	BCS-OVA	CSOVA	SECOC-OVA
1＃	3.00	**2.70**	5	4.50	**3.60**	5
2＃	**13.10**	**13.10**	25	**15.30**	18.40	25
3＃	**25.00**	**25.00**	50	**26.80**	31.60	50
4＃	**54.50**	**54.50**	81.70	**58.10**	68.20	100
5＃	**85.60**	87.60	174	**99.30**	122.30	150
6＃	**109.20**	**109.20**	194	**130.20**	138.7	200
Average cost	**48.40**	48.68	88.28	**55.70**	63.80	88.33

表 4 不同方法在人工数据集上的误分类总代价(第二种类型多类代价矩阵)

Table 4 Test cost of different methods on artificial datasets using cost matrix in type b

Datasets	BCS-OVO	CSOVO	SECOC-OVO	BCS-OVA	CSOVA	SECOC-OVA
1＃	**24.50**	**24.50**	40.70	19.70	**15.00**	47
2＃	**100.50**	**100.50**	301.30	**106.70**	124.30	265
3＃	**180.90**	**180.90**	616	**228.00**	234.40	590
4＃	216.10	**215.70**	807.80	264.60	**248.40**	740
5＃	**446.30**	**446.30**	785.20	782.30	**671.50**	1200
6＃	**620.20**	621.70	1173.10	780.60	**758.70**	1720
Average cost	264.75	264.93	620.68	363.65	**342.05**	760.33

1) http://www.csie.ntu.edu.tw/~htlin/program/CSSVM

表 5 不同方法在人工数据集上的误分类总代价(第三种类型多类代价矩阵)

Table 5 Test cost of different methods on artificial datasets using cost matrix in type c

Datasets	BCS-OVO	CSOVO	SECOC-OVO	BCS-OVA	CSOVA	SECOC-OVA
1#	31.80	**27.00**	67.40	38.60	**33.80**	83
2#	**102.60**	107.20	267.90	150.30	**104.50**	305
3#	286.20	**255.10**	533.10	453.20	**341.40**	600
4#	560	**447.90**	855	721.80	**563.20**	1100
5#	773.90	**543**	1541.50	939.70	**721.50**	1560
6#	863.60	**768.10**	2379.90	1327.20	**1115.40**	2720
Average cost	436.35	**358.05**	940.80	605.13	**479.96**	1061.30

表 3－表 5 中每个数据集上 3 种方法中的最小误分类总代价被加粗显示。每张表的最后一行给出了每个方法在所有数据集上的误分类总代价的平均值。

1)由表 3 和表 4 可以看出，BCS-OVO 和 BCS-OVA 拥有与 CSOVO 和 CSOVA 相似的误分类总代价的平均值。但在表 6 中，BCS-OVO 和 BCS-OVA 误分类总代价的平均值明显高于 CSOVO 和 CSOVA。即对于前两种类型的多类代价矩阵，本文方法的性能与 CSOVO 和 CSOVA 的性能相当甚至更好，但是在第三种类型的多类代价矩阵上性能较差。

2)由表 3－表 5 可以看出，本文提出方法的误分类总代价的平均值都显著低于 SECOC-OVO 和 SECOC-OVA 的平均值，性能明显更优。

(2)UCI 数据集

表 6－表 8 分别给出了 3 种不同类型的多类代价矩阵下，不同方法在 UCI 数据集上的误分类总代价。表中每个数据集上 3 种方法中的最小误分类总代价被加粗显示。每张表的最后一行给出了每个方法在所有数据集上的误分类总代价的平均值。

表 6 不同方法在 UCI 数据集上的误分类总代价(第一种类型多类代价矩阵)

Table 6 Test cost of different methods on UCI datasets using cost matrix in type a

Datasets	BCS-OVO	CSOVO	SECOC-OVO	BCS-OVA	CSOVA	SECOC-OVA
balance	**8.70**	11.90	57.70	**11.20**	11.30	57.60
glass	**3.40**	**3.40**	12.60	**4.20**	**4.20**	14.40
iris	**4**	10.50	6.10	**1.10**	**1.10**	10
letter	39	**33**	117.10	64.50	**59.40**	116.60
page-blocks	**27.80**	79.30	57.40	**32.80**	37.60	56
sat	118.60	**116.40**	489.80	123.40	**123.20**	490.20
segmentation	14.70	**13.20**	254.20	**175**	283.20	2640
soybean	17.50	**17.30**	30.20	**19**	**19**	29.60
thyroid	2.10	**1**	4.60	1.40	1.40	6.50
vehicle	32.90	**30.90**	79.50	30.40	**29.50**	64.70
vowel	**2.40**	2.50	98.90	4.70	**4.30**	90
wine	**2.10**	2.60	11.20	**3.30**	**3.30**	11.90
yeast	**62.50**	64.10	148.10	**66**	75.90	124
zoo	**2.70**	2.70	6	**2.90**	**2.90**	6
Average cost	**24.17**	27.77	98.10	**38.56**	46.87	265.53

表 7 不同方法在 UCI 数据集上的误分类总代价(第二种类型多类代价矩阵)

Table 7 Test cost of different methods on UCI datasets using cost matrix in type b

Datasets	BCS-OVO	CSOVO	SECOC-OVO	BCS-OVA	CSOVA	SECOC-OVA
balance	**31.90**	**31.90**	292.90	**30.70**	45.70	720
glass	**32.10**	**32.10**	197.50	**51.50**	51.60	205.30
iris	**2.50**	**2.50**	6.90	4.70	**2.70**	55
letter	350.40	**348.10**	1342.90	**705.50**	**705.50**	1342.90
page-blocks	**175.50**	301.60	3451.50	**152.70**	170.80	447
sat	**605.50**	**605.50**	5707.90	863.30	**729**	5825.10
segmentation	119.40	**108.10**	2541	192.40	**159.40**	2277
soybean	196.30	**190.50**	310.20	**205.30**	**205.30**	310.20
thyroid	**2.50**	**2.50**	22.10	2.80	**2.40**	30
vehicle	**165.80**	**165.80**	197.90	247.80	**187.40**	534.10
vowel	**9**	**9**	819	**17.50**	**17.50**	819
wine	**16.70**	**16.70**	148.20	**33.90**	**33.90**	132.60
yeast	569.20	**557.10**	940.50	653.90	**533.90**	940.50
zoo	21.50	**18.50**	83	**22.30**	**22.30**	39.80
Average cost	**164.16**	170.70	1147.30	227.45	**204.81**	977.03

表 8 不同方法在 UCI 数据集上的误分类总代价(第三种类型多类代价矩阵)

Table 8 Test cost of different methods on UCI datasets using cost matrix in type c

Datasets	BCS-OVO	CSOVO	SECOC-OVO	BCS-OVA	CSOVA	SECOC-OVA
balance	29.50	**20.40**	99.90	71.90	**40.60**	259.20
glass	29.40	**22.30**	105	**33.70**	34	105
iris	5.60	**4.80**	21.90	**6.40**	6.90	25
letter	309.10	**241.30**	1026.20	**541.50**	**541.50**	1026.20
page-blocks	498.30	**257.40**	746.50	**239.60**	299.80	735.70
sat	797.30	**366.40**	4003.10	763.10	**746.60**	4655.70
segmentation	**122.10**	150.10	1222.70	**254.30**	272.90	1914
soybean	196.40	**132.70**	321.80	**209.40**	**209.40**	321.80
thyroid	7	**6.50**	32.50	**7.40**	7.60	42.50
vehicle	277.10	**189.60**	767	333.70	**319.30**	991.40
vowel	6.30	**5.60**	864	**26.50**	**26.50**	864
wine	**17.30**	24.10	77.60	**34.40**	**34.40**	133.60
yeast	753.70	**623**	768.90	**646.20**	672.40	770.90
zoo	25.60	**16.30**	50.10	**25.10**	**25.10**	59.10
Average cost	219.62	**147.17**	721.94	**228.08**	231.21	850.29

1)本文方法在3种类型的多类代价矩阵下取得了与CSOVO和CSOVA相似的结果。除了第三种类型的多类代价矩阵下,BCS-OVO的误分类总代价的平均值要高于CSOVO的平均值,这一结论与在人工数据集上的结论是一致的。

2)本文方法与SECOC-OVO和SECOC-OVA相比性能明显更好。同时,实验也验证了在实际问题中利用类内类间距离计算类间的可分性是可行的,因为对于实际应用而言计算某一类的概率密度函数还是相当困难的。

通过以上实验可知:本文提出的基于多类代价矩阵计算基分类器的二类代价矩阵的方法是有效的、可行的,且基于此提出的新的二类代价敏感纠错输出编码BCS_ECOC与现有方法相比具有相似甚至更好的分类性能。

结束语 本文研究了基于纠错输出编码实现多类代价敏感分类的方法,提出一种新的代价敏感纠错输出编码框架;将多类代价敏感分类问题分解为多个二类代价敏感分类问题,为获得其中每个二类代价敏感基分类器的二类代价矩阵,提出了利用已知多类代价矩阵计算每一类的误分类代价的期望值的方法,并给出了为每个基分类器计算二类代价矩阵的通用的计算公式;为验证提出方法的性能,将其与现有方法在不同数据集上进行了比较,实验结果表明,提出方法具有相似甚至更好的性能。相比于CSOVO和CSOVA等特定的方法,BCS_ECOC作为一种通用框架,可以应用现有的多种编码方法和解码方法,从而为提高多类代价敏感分类的性能提供了可能。

参 考 文 献

[1] ALI S, MAJID A, JAVED S G, et al. Can-CSC-GBE: Developing Cost-sensitive Classifier with Gentleboost Ensemble for breast cancer classification using protein amino acids and imbalanced data[J]. Computers In Biology and Medicine, 2016, 73: 38-46.

[2] KIM Y J, BAIK B, CHO S. Detecting financial misstatements with fraud intention using multi-class cost-sensitive learning[J]. Expert Systems With Applications, 2016, 62: 32-43.

[3] ZHOU Z H, LIU X Y. On multi-class cost-sensitive learning [J]. Computational Intelligence, 2010, 26(3): 232-257.

[4] KULLUK S, ÖZBAKIR L, TAPKAN P Z, et al. Cost-sensitive meta-learning classifiers: MEPAR-miner and DIFACONN-miner [J]. Knowledge-Based Systems, 2016, 98: 148-161.

[5] ZHANG G Q, SUN H J, et al. Cost-sensitive dictionary learning for face recognition[J]. Pattern Recognition, 2016, 60: 613-629.

[6] JIANG L X. Cost-sensitive Bayesian network classifiers[J]. Pattern Recognition Letters, 2014, 45: 211-216.

[7] BÁÑEZ A, BIELZA C, LARRAÑAGA P. Cost-sensitive selective naïve Bayes classifiers for predicting the increase of the h-index for scientific journals[J]. Neurocomputing, 2014, 135: 42-52.

[8] CHEN Z, XIAO X Y, LI C S, et al. Real-time transient stability status prediction using cost-sensitive extreme learning machine [J]. Neural Computing and Application, 2016, 27: 321-331.

[9] ZHANG L, ZHANG D. Evolutionary Cost-Sensitive Extreme Learning Machine[J]. IEEE Transactions on Nueral Networks and Learning Systems, 2017, 28(12): 3045-3060.

[10] DIETTERICH T, BAKIRI G. Solving multiclass learning problems via error-correcting output codes[J]. Journal of Artificial Intelligence Research, 1995, 34(2): 263-268.

[11] LEI L, WANG X D. An Overview of Multi-Classification Based on Error-Correcting Output Codes[J]. Acta Electronica Sinica, 2014, 42(9): 1794-1800.

[12] BAI X L, NIWAS S I, LIN W S, et al. Learning ECOC code matrix for multiclass classification with application to Glaucoma diagnosis[J]. Journal of Medical Systems, 2016, 40: 78.

[13] LIU K H, ZENG Z H, NG V T Y. A hierarchical ensemble of ECOC for cancer classification based on multi-class microarray data[J]. Information Sciences, 2016, 349-350: 102-118.

[14] SANTHANAM V, MORARIU V I, HARWOOD D, et al. A nonparametricapproach to extending generic binary classifiersfor multi-classification[J]. Pattern Recognition, 2016, 58(1): 149-158.

[15] LIN H T. From ordinal ranking to binary classification[D]. California Institute of Technology, 2008.

[16] LANGFORD J, BEYGELZIMER A. Sensitive error correcting output codes[J]. Lecture Note in Artificial Intelligence, 2005, 3559(1): 158-172.

[17] LIN H T. Reduction from cost-sensitive multiclass classification to one-versus-one binary classification[C]// JMLR: Workshop and Conference Proceedings, 2014, 39: 371-386.

[18] ASUNCION A, NEWMAN D. School of Information and Computer Sciences[M]// UCI machine learning repository. Irvine, CA, USA: University of California, 2007.

[19] BEYGELZIMER A, LANGFORD J, RAVIKUMAR P. Error correcting tournaments[EB/OL]. http://arxiv.org/abs/0902.3176.

[20] TING K M. An instance-weighting method to induce cost-sensitive trees[J]. IEEE Transactions on Knowledge and Data Engineering, 2002, 14(3): 659-665.

[21] ZHANG L, ZHANG D. Evolutionary Cost-Sensitive Extreme Learning Machine[J]. IEEE Transactions on Nueral Networks and learning Systems, 2017, 28(12): 3045-3060.

WU Chong-ming, born in 1966, Ph.D, associate professor. Hismain research interests include machine learning and intelligent information processing.

WANG Xiao-dan, born in 1966, Ph.D, professor. Her research interests include machine learning, pattern recognition.